# TROPICAL RAINFALL MEASURING MISSION PRECIPITATION PROCESSING SYSTEM

## File Specification
## 2A12

### Version 7

September 9, 2013

## 0.1 2A12 - TMI Profiling

2A-12, "TMI Profiling", generates surface rainfall and vertical hydrometeor profiles on a pixel by pixel basis from the TRMM Microwave Imager (TMI) brightness temperature data using the Goddard Profiling algorithm GPROF2010. Because the vertical information comes from a radiometer, it is not written out in independent vertical layers like the TRMM Precipitation Radar. Instead, the output is referenced to one of 100 typical structures for each hydrometeor or heating profile. These vertical structures are referenced as clusters in the output structure. Vertical hydrometeor profiles can be reconstructed to 28 layers by knowing the cluster number (i.e. shape) of the profile and a scale factor that is written for each pixel.

GPROF 2010 (2A12, V7) vs GPROF 2004 (2A12, V6)
(A short explanation of the differences)

The basic change from 2A12, V6 (corresponding to GRPOF2004) and 2A12, V7 (Corresponding to GPROF2010) is the use of the observed databases instead of CRM databases. GPROF 2004 used a set of eight cloud resolving model simulations to create the a-priori databases for the Bayesian retrieval scheme. This methodology had issues over oceans due primarily to a lack of completeness or representativeness of this database. GPROF 2010 replaced these simulations with combined radar/radiometer/CRM retrievals that are supposed to better represent the actual profiles seen over oceans. One year of combined retrievals (detailed in Kummerow et al., J. Atmos. and Oceanic Tech., 2011) are used to construct the a-priori database. Note that while the a-priori database in GPROF2004 was stratified by SST, the GPROF2010 database is stratified by both SST and total precipitable water (TPW). The retrieval was also changed over land. Here, new regressions have been established to match PR rainfall. These are done separately for convective and stratiform precipitation. The regressions are global. Changes to the algorithm led to some natural changes in the output format. Below is a basic description of the new or modified fields.

**pixelStatus** flags any pixel that was eliminated due to QC screens. The value itself, if not equal to zero, contains information about the specific QC procedure that identified the pixel as bad. Values are available below. If PixelStatus is not equal to zero, all other fields are set to missing.

**qualityFlag** is a qualitative flag in GPROF2010 to guide users in areas where GPROF 2010 has retrieved rain but with less than optimal confidence. QualityFlag = 0 means highest confidence. QualityFlag = 1 is used to identify pixels that are probably good but climate trends and very small signals should not be taken from these. Pixels with a small sun glint angle, for instance, are identified in this category. QualityFlag = 2 should be used primarily for qualitative purposes. This category, for instance, identifies areas where GPROF databases were searched far beyond the appropriate SST and TPW bin before a solution could be found.

**surfacePrecipitation** is the precipitation rate corresponding to each pixel. This value should be accumulated if area or temporal averages are sought. SurfaceRain represents the liquid component of SurfacePrecipitation.

**New/Additional diagnostic parameters over ocean**

**oceanSearchRadius** contains the specific value of how far the algorithm searched the a-priori database beyond the nominal SST and TPW bin on the pixel. This field is intended for users willing to parse the QualityFlag further and then use certain subsets of the solution.

**chiSquared** (ocean only): In addition to precipitation, GPROF2010 also retrieves and outputs surface wind speed, water vapor and cloud water. The retrieval uses an optimal estimation methodology (similar to 1D Var). ChiSquared is the diagnostic that measures the fit of the non-raining solution. A ChiSquared value of less than 5 is equivalent to a quality flag of 0, implying that the wind, water vapor and cloud water are good. ChiSquared values between 5 and 10 are generally trustworthy but they can be contaminated with light rain and should probably not be used for studying small trends or correlations. Values above 10 are equivalent to qualityFlag = 2 and should not be used for non-raining pixels. Precipitation can be retrieved simultaneously with these parameters. A typical scenario is one where chiSquared is less than 5 for no rain or very light (less than 0.1 mm/hr) rain, and increases as the rainfall increases. For significant rain, chiSquared is often very high, while qualityFlag = 0 – simply stating that background fields cannot be retrieved reliably in heavy rainfall conditions.

**probabilityOfPrecip** (ocean only): Because the ocean database now follows PR/TMI observations, including the proper ratio of raining and non-raining pixels, the retrieval no longer screens for raining pixels before applying the Bayesian scheme. All pixels are compared to the a-priori database. This leads to an unexpected, but correct result that nearly all pixels have some very small probability of rainfall as well as rainfall rate. This occurs because a given set of observed Tbs has similarities to one or more entry in the database that has some rain, even if the majority of matches are to non-raining pixels. To allow the user to determine if a single pixel is likely raining or not, GPROF 2010 writes out the probability of precipitation in addition to the mean rainfall rate. Using a threshold of 50% for the probability of rain was found to be a fairly good indicator of rain in the FOV when compared to PR (at the lower TMI resolution). For monthly accumulations over oceans, the user should count all pixels (i.e. use surfacePrecipitation). For making instantaneous rainfall maps, or maps of the probability of rain, higher probabilityOfPrecip threshold should be used or maps will reflect all pixels in which rain is possible (instead of likely). A probability of precipitation of 50% seems to track raining areas pretty well but other thresholds can be chosen as well. The transition between 0% rain and 100% rain is quite steep so that the exact threshold for the probability of precipitation is not critical.

**Other Changes to Output**

The final major change over oceans deals with the format of the output structure. GPROF2004 (2A12, V6) produced output in 14 layers for rain water, cloud water, precipitation ice, cloud ice and latent heating. GPROF 2010 uses 28 layers but does not write these out explicitly. Instead, it uses only 100 typical profiles (clusters) for each hydrometeor (or LH) species. The profile shapes for each cluster are given in the DataHeader while the pixel carries only the profile cluster number and the scaling factor for each hydrometeor type. Freezing levels are not mixed so the freezing level index is also given for each pixel. This reduces the size of the files significantly although the profiles have to be reconstructed from the shape and scale information before it can be used.

Land retrievals use a regression against PR data as described above. The surface parameters (wind, water vapor and cloud water) are not retrieved over land. As such, chi squared does not exist. Land still employs a screen to determine if it is raining or not. Unfortunately, the screens are not always unambiguous and an ambiguousFlag is therefore defined. If a pixel is defined as Ambiguous, it is given a low qualityFlag of 2. Ambiguous pixels should be avoided by all except expert users who know each of the ambiguous screens. Because the ambiguous definition does not translate into a probability of precipitation as it does over ocean, the probabilityOfPrecip is left as missing over land and coast. Land and Coast retrievals do not retrieve any structure for TMI. All vertical information is also set to missing at this time.

The format of this product was designed in consultation with the TMI algorithm scientists. The following sections describe the structure and contents of the format.

Dimension definitions:

| | | |
|---|---|---|
| nscan | var | Number of scans in the granule. |
| npixel | 208 | Number of pixels in each scan. |
| nspecies | 6 | Number of hydrometeor species. Species are: 1=cloud liquid water content ($g/m^3$), 2=rain water content ($g/m^3$), 3=cloud ice water content ($g/m^3$), 4=snow water content ($g/m^3$), 5=graupel water content ($g/m^3$), 6=latent heating (K/h). |
| nfindex | 13 | Number of freezing height indeces. |
| nlayer | 28 | Number of profiling layers. The top height of each layer is specified in heightLayerTop. |
| ncluster | 100 | Number of clusters at each freezing height. |

Figure 1 through Figure 6 show the structure of this product. The text below describes the contents of objects in the structure, the C Structure Header File and the Fortran Structure Header File.

**FileHeader** (Metadata):
FileHeader contains general metadata. This group appears in all data products. See Metadata for TRMM Products for details.

**InputRecord** (Metadata):
InputRecord contains a record of input files for this granule. This group appears in Level 1 and Level 2 data products. Level 3 time averaged products have the same information separated into 3 groups since they have many inputs. See Metadata for TRMM Products for details.

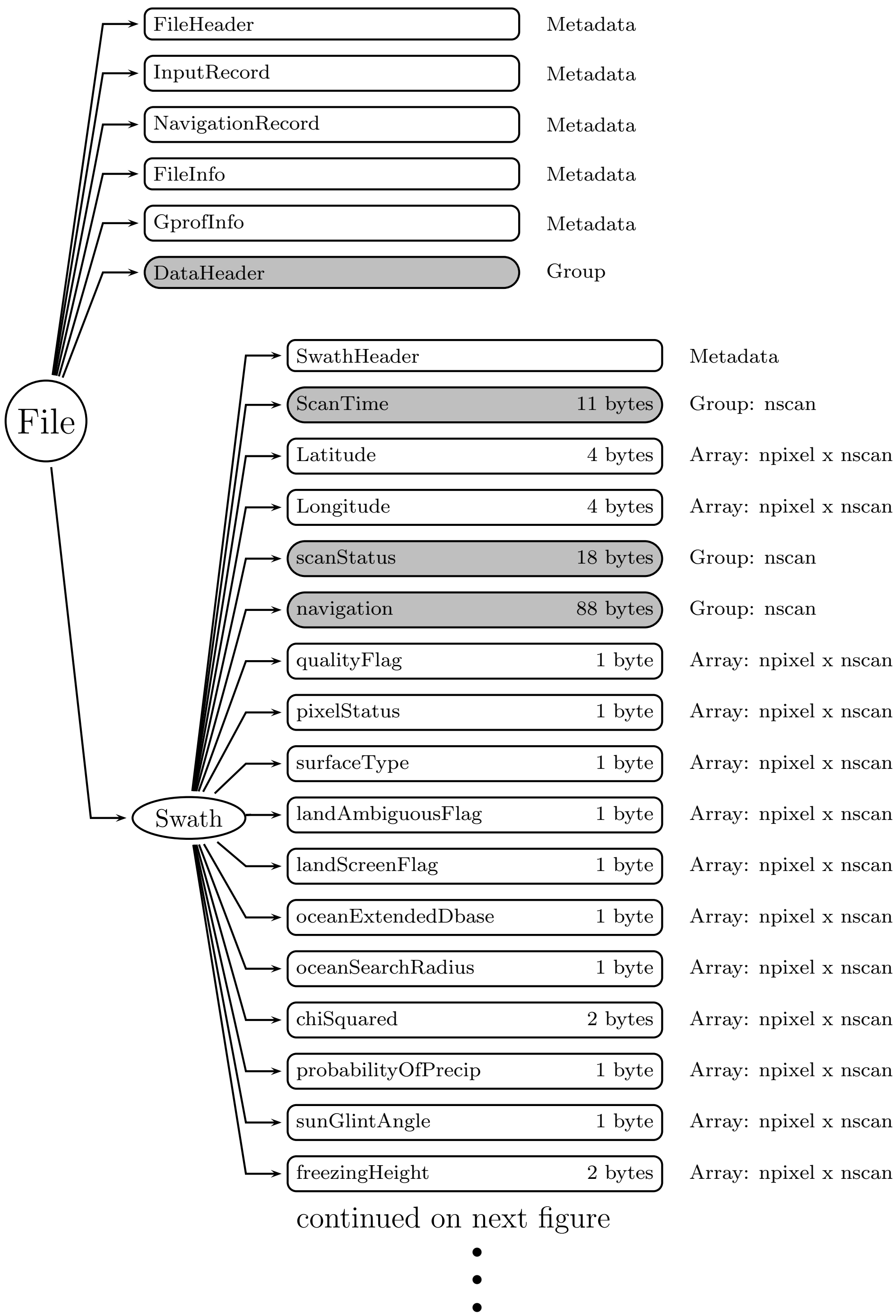


Figure 1: Data Format Structure for 2A12, TMI Profiling

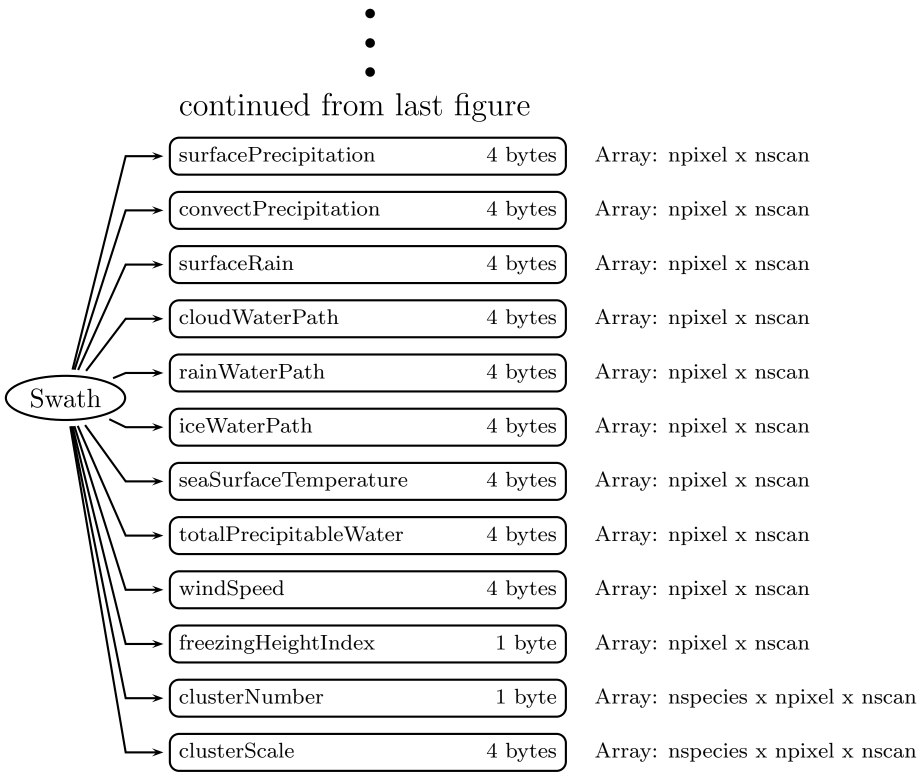


Figure 2: Data Format Structure for 2A12, TMI Profiling

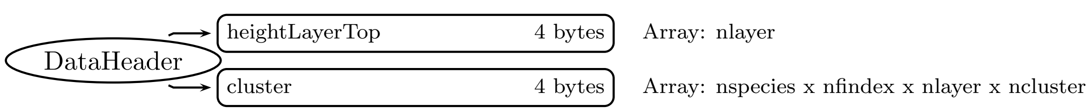


Figure 3: Data Format Structure for 2A12, DataHeader

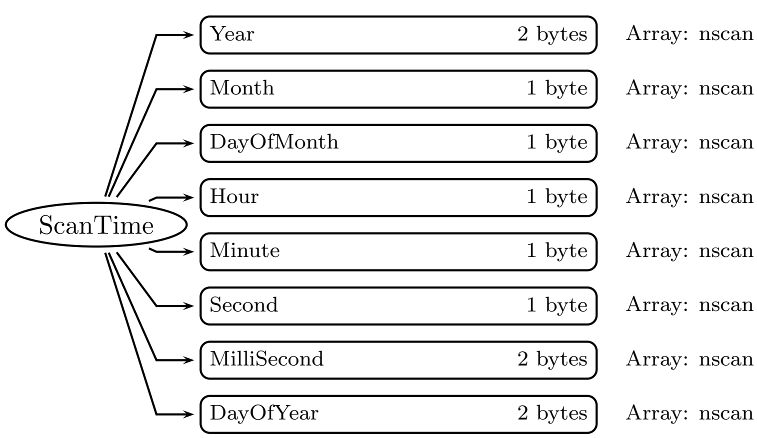


Figure 4: Data Format Structure for 2A12, ScanTime

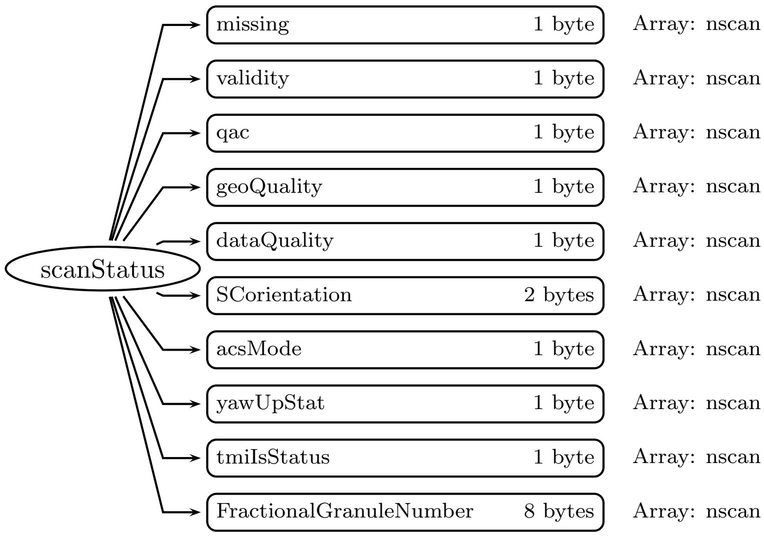


Figure 5: Data Format Structure for 2A12, scanStatus

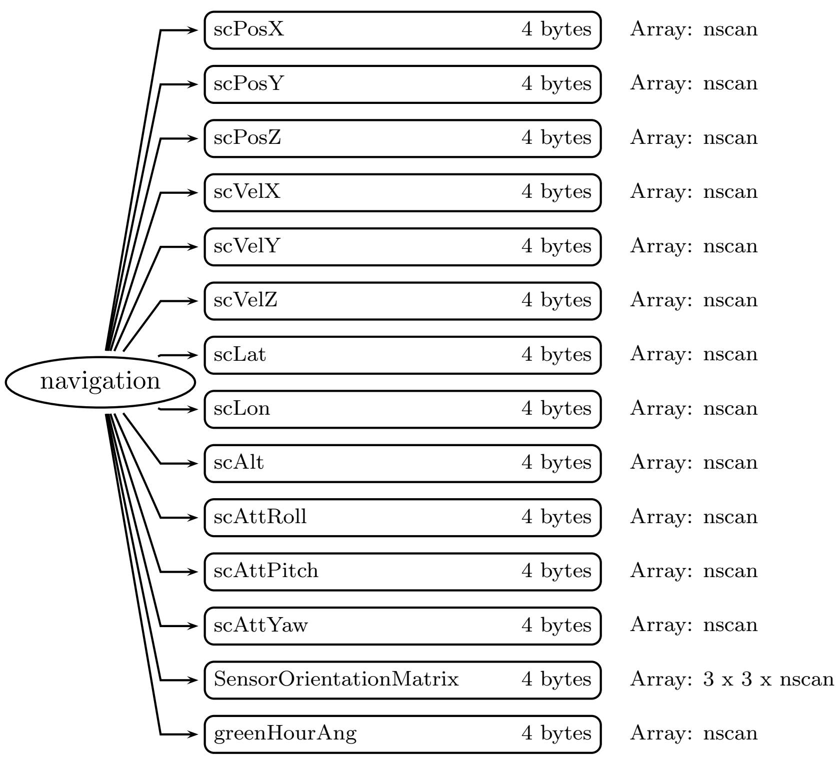


Figure 6: Data Format Structure for 2A12, navigation

**NavigationRecord** (Metadata):
NavigationRecord contains navigation metadata for this granule. This group appears in Level 1 and Level 2 data products. See Metadata for TRMM Products for details.

**FileInfo** (Metadata):
FileInfo contains metadata used by the PPS I/O Toolkit (TKIO). This group appears in all data products. See Metadata for TRMM Products for details.

**GprofInfo** (Metadata):
GprofInfo contains metadata required by Gprof. Used by 2A12 only. See Metadata for TRMM Products for details.

## **DataHeader** (Group)

**heightLayerTop** (4-byte float, array size: nlayer):
Height of the top of each atmospheric layer in the rain profile. Values range from 0 to 18.0 km. Special values are defined as:
-9999.9 Missing value

**cluster** (4-byte float, array size: nspecies x nfindex x nlayer x ncluster):
Hydrometeor profile shapes. Dimensions are hydrometeor/heating species (6) consisting of cloud water, rain water, cloud ice, snow, graupel, and latent heating; freezing height index (13) for freezing levels starting at 250m and going to 4.5 km in 250 m intervals, vertical layers (28) starting at 500 m and going to 18 km at 500 m layers in the lower troposphere and 1 km in the upper troposphere; and cluster number (currently 100 shapes). To recover values in a profile see the description below the variable clusterScale.

Special values are defined as:
-9999.9 Missing value

## **Swath** (Swath)

**SwathHeader** (Metadata):
SwathHeader contains metadata for swaths. This group appears in Level 1 and Level 2 data products. See Metadata for TRMM Products for details.

## **ScanTime** (Group)

**Year** (2-byte integer, array size: nscan):
4-digit year, e.g., 1998. Values range from 1950 to 2100 years. Special values are defined as:
-9999 Missing value

**Month** (1-byte integer, array size: nscan):
Month of the year. Values range from 1 to 12 months. Special values are defined as:
-99 Missing value

**DayOfMonth** (1-byte integer, array size: nscan):
Day of the month. Values range from 1 to 31 days. Special values are defined as:
-99 Missing value

**Hour** (1-byte integer, array size: nscan):
UTC hour of the day. Values range from 0 to 23 hours. Special values are defined as:
-99 Missing value

**Minute** (1-byte integer, array size: nscan):
Minute of the hour. Values range from 0 to 59 minutes. Special values are defined as:
-99 Missing value

**Second** (1-byte integer, array size: nscan):
Second of the minute. Values range from 0 to 60 s. Special values are defined as:
-99 Missing value

**MilliSecond** (2-byte integer, array size: nscan):
Thousandths of the second. Values range from 0 to 999 ms. Special values are defined as:
-9999 Missing value

**DayOfYear** (2-byte integer, array size: nscan):
Day of the year. Values range from 1 to 366 days. Special values are defined as:
-9999 Missing value

**Latitude** (4-byte float, array size: npixel x nscan):
The earth latitude of the center of the IFOV at the altitude of the earth ellipsiod. Latitude is positive north, negative south. Values range from -90 to 90 degrees. Special values are defined as:
-9999.9 Missing value

**Longitude** (4-byte float, array size: npixel x nscan):
The earth longitude of the center of the IFOV at the altitude of the earth ellipsiod. Longitude is positive east, negative west. A point on the 180th meridian has the value -180 degrees. Values range from -180 to 180 degrees. Special values are defined as:
-9999.9 Missing value

## **scanStatus** (Group)

**missing** (1-byte integer, array size: nscan):
Missing indicates whether information is contained in the scan data. The values are:

```
0 Scan data elements contain information
1 Scan was missing in the telemetry data
```

**validity** (1-byte integer, array size: nscan):
Validity is a summary of status modes. If all status modes are routine, all bits in Validity = 0. Routine means that scan data has been measured in the normal operational situation as far as the status modes are concerned. Validity does not assess data or geolocation quality. Validity is broken into 8 bit flags. Each bit = 0 if the status is routine but the

bit = 1 if the status is not routine. Bit 0 is the least significant bit (i.e., if bit i = 1 and other bits = 0, the unsigned integer value is 2**i). The non-routine situations follow:

```
Bit Meaning if bit = 1
0   Spare (always 0)
1   Non-routine spacecraft orientation (2 or 3)
2   Non-routine ACS mode (other than 4)
3   Non-routine yaw update status (0 or 1)
4   Non-routine instrument status (Bit 0 = 0 or bit 1 = 0)
5   Non-routine QAC (non-zero)

6   21 GHz Cold Count Flag (1 if Flag set)
7   Spare (always 0)
```

**qac** (1-byte integer, array size: nscan):
The Quality and Accounting Capsule of the Science packet as it appears in Level-0 data. If no QAC is given in Level-0, which means no decoding errors occurred, QAC in this format has a value of zero.

**geoQuality** (1-byte integer, array size: nscan):
geoQuality is broken into 8 one-bit flags. Some flags represent problems but other flags are informational. Bits 0, 5, and 6 represent problems: 0 = 'good' quality and 1 = 'bad' quality. It is recommended not to use scans when any problem flag is 1. The informational flags have: 0 = routine conditions and 1 = non-routine conditions. Bit 0 is the most significant bit (i.e., if bit i = 1 and other bits = 0, the unsigned integer value is 2**(7-i). Note that good scans may have non-zero geoQuality. Each flag is listed below.

```
Bit Meaning if bit = 1
0   Grossly bad geolocation results:
    Spacecraft position vector magnitude outside range 6715 to 6790 km.
    Z component of midpoint of scan outside range -4100 to 4100 km.
    Distance from S/C to midpoint of scan outside range 500 to 750 km.
1   Unexpectedly large scan to scan jumps in geolocated positions in along and
    cross track directions for first, middle, and last pixels in each scan.
    Allowed deviation from nominal jump in along track motion = 3.0 km (first
    pixel), 3.0 km (middle pixel), and 3.0 km (last pixel).
    Allowed deviation from nominal jump in cross track motion = 3.0 km (first
    pixel), 3.0 km (middle pixel), and 3.0 km (last pixel).
    Bit set in normal mode only.
2   Scan to scan jumps in yaw, pitch, and roll exceed maximum values.  Values
    are :  yaw = 0.005 radians; pitch = 0.005 radians; roll = 0.005 radians.
    Bit set in normal control mode only.
3   In normal mode, yaw outside range (-0.005, 0.005) radians; pitch outside
    range (-0.005, 0.005) radians; roll outside range (-0.005, 0.005) radians.
```

```
4   Satellite undergoing maneuvers during which geolocation will be less
    accurate.
5   Summary QA flag for dataQuality:  Set to 1 if bit 0 is 1 or bit 6 is 1,
    i.e. Grossly bad or failed geolocation calculations.
    Science data use not recommended.
6   Geolocation calculations failed (fill values inserted in the per pixel
    geolocation products, but not in metadata).
7   Missing attitude data.  ACS data gap larger than 1.0 seconds.
      Pitch, roll, and yaw are interpolated or extrapolated from nearby data.
```

**dataQuality** (1-byte integer, array size: nscan):
dataQuality is a flag for overall scan quality. Unless this is 0, the scan data is meaningless to higher science processing. Bit 0 is the least significant bit (i.e., if bit i = 1 and other bits = 0, the unsigned integer value is 2**i).

```
Bit  Meaning if bit = 1
0    missing
5    geoQuality indicates bad or missing values

6    validity bits 0-5 not all normal
```

**SCorientation** (2-byte integer, array size: nscan):
The positive angle of the spacecraft vector (v) from the satellite forward direction of motion, measured clockwise facing down. We define v in the same direction as the spacecraft axis +X, which is also the center of the TMI scan. If +X is forward, SCorientation is 0. If -X is forward, SCorientation is 180. If -Y is forward, SCorientation is 90. Values range from 0 to 360 degrees. Special values are defined as:

| | |
|---|---|
| -8003 | Inertial |
| -8004 | Unknown |
| -9999 | Missing value |

**acsMode** (1-byte integer, array size: nscan):

```
Value Meaning
0     Standby
1     Sun Acquire
2     Earth Acquire
3     Yaw Acquire
4     Nominal
5     Yaw Maneuver
6     Delta-H (Thruster)
7     Delta-V (Thruster)
8     CERES Calibration
```

**yawUpStat** (1-byte integer, array size: nscan):

```
Value Meaning
0     Inaccurate
1     Indeterminate
2     Accurate
```

**tmiIsStatus** (1-byte integer, array size: nscan):
Bit 0 is the most significant bit (i.e., if bit i = 1 and other bits = 0, the unsigned integer value is 2**(8-i) - 1).

```
Bit Meaning
00  Receiver Status (1=ON, 0=OFF)
01  Spin-up Status (1=ON, 0=OFF)
02  Spare Command 1 Status
03  Spare Command 2 Status
04  1 Hz Clock Select (1=A, 0=B)

05  Spare
06  Spare Command 4 Status
07  Spare Command 5 Status
```

**FractionalGranuleNumber** (8-byte float, array size: nscan):
The floating point granule number. The granule begins at the Southern-most point of the spacecraft's trajectory. For example, FractionalGranuleNumber = 10.5 means the spacecraft is halfway through granule 10 and starting the descending half of the granule. Values range from 0 to 100000. Special values are defined as:
-9999.9 Missing value

## navigation (Group)

**scPosX** (4-byte float, array size: nscan):
The x component of the position (m) of the spacecraft in Geocentric Inertial Coordinates at the Scan mid-Time (i.e., time at the middle pixel/IFOV of the active scan period). Geocentric Inertial Coordinates are also commonly known as Earth Centered Inertial coordinates. These coordinates will be True of Date (rather than Epoch 2000 which are also commonly used), as interpolated from the data in the Flight Dynamics Facility ephemeris files generated for TRMM.

**scPosY** (4-byte float, array size: nscan):
The y component of the position (m) of the spacecraft in Geocentric Inertial Coordinates. See scPosX.

**scPosZ** (4-byte float, array size: nscan):
The z component of the position (m) of the spacecraft in Geocentric Inertial Coordinates. See scPosX.

**scVelX** (4-byte float, array size: nscan):
The x component of the velocity ($ms^{-1}$) of the spacecraft in Geocentric Inertial Coordinates at the Scan mid-Time.

**scVelY** (4-byte float, array size: nscan):
The y component of the velocity ($ms^{-1}$) of the spacecraft in Geocentric Inertial Coordinates at the Scan mid-Time.

**scVelZ** (4-byte float, array size: nscan):
The z component of the velocity ($ms^{-1}$) of the spacecraft in Geocentric Inertial Coordinates at the Scan mid-Time.

**scLat** (4-byte float, array size: nscan):
The geodedic latitude (decimal degrees) of the spacecraft at the Scan mid-Time.

**scLon** (4-byte float, array size: nscan):
The geodedic longitude (decimal degrees) of the spacecraft at the Scan mid-Time.

**scAlt** (4-byte float, array size: nscan):
The altitude (m) of the spacecraft above the Earth Ellipsiod at the Scan mid-Time.

**scAttRoll** (4-byte float, array size: nscan):
The satellite attitude Euler roll angle (degrees) at the Scan mid-Time. The order of the components in the file is roll, pitch, and yaw. However, the angles are computed using a 3-2-1 Euler rotation sequence representing the rotation order yaw, pitch, and roll for the rotation from Orbital Coordinates to the spacecraft body coordinates. Orbital Coordinates represent an orthogonal triad in Geocentric Inertial Coordinates where the Z-axis is toward the geocentric nadir, the Y-axis is perpendicular to the spacecraft velocity opposite the orbit normal direction, and the X-axis is approximately in the velocity direction for a near circular orbit. Note this is geocentric, not geodetic, referenced, so that pitch and roll will have twice orbital frequency components due to the onboard control system following the oblate geodetic Earth horizon. Note also that the yaw value will show an orbital frequency component relative to the Earth fixed ground track due to the Earth rotation relative to inertial coordinates.

**scAttPitch** (4-byte float, array size: nscan):
The satellite attitude Euler pitch angle (degrees) at the Scan mid-Time. The order of the components in the file is roll, pitch, and yaw. However, the angles are computed using a 3-2-1 Euler rotation sequence representing the rotation order yaw, pitch, and roll for the rotation from Orbital Coordinates to the spacecraft body coordinates. Orbital Coordinates represent an orthogonal triad in Geocentric Inertial Coordinates where the Z-axis is toward the geocentric nadir, the Y-axis is perpendicular to the spacecraft velocity opposite the orbit normal direction, and the X-axis is approximately in the velocity direction for a near circular orbit. Note this is geocentric, not geodetic, referenced, so that pitch and roll will have twice orbital frequency components due to the onboard control system

following the oblate geodetic Earth horizon. Note also that the yaw value will show an orbital frequency component relative to the Earth fixed ground track due to the Earth rotation relative to inertial coordinates.

**scAttYaw** (4-byte float, array size: nscan):
The satellite attitude Euler yaw angle (degrees) at the Scan mid-Time. The order of the components in the file is roll, pitch, and yaw. However, the angles are computed using a 3-2-1 Euler rotation sequence representing the rotation order yaw, pitch, and roll for the rotation from Orbital Coordinates to the spacecraft body coordinates. Orbital Coordinates represent an orthogonal triad in Geocentric Inertial Coordinates where the Z-axis is toward the geocentric nadir, the Y-axis is perpendicular to the spacecraft velocity opposite the orbit normal direction, and the X-axis is approximately in the velocity direction for a near circular orbit. Note this is geocentric, not geodetic, referenced, so that pitch and roll will have twice orbital frequency components due to the onboard control system following the oblate geodetic Earth horizon. Note also that the yaw value will show an orbital frequency component relative to the Earth fixed ground track due to the Earth rotation relative to inertial coordinates.

**SensorOrientationMatrix** (4-byte float, array size: 3 x 3 x nscan):
SensorOrientationMatrix is the rotation matrix from the instrument coordinate frame to Geocentric Inertial Coordinates at the Scan mid-Time. It is unitless.

**greenHourAng** (4-byte float, array size: nscan):
The rotation angle (degrees) from Geocentric Inertial Coordinates to Earth Fixed Coordinates.

**qualityFlag** (1-byte integer, array size: npixel x nscan):
qualityFlag indicates a generalized quality of the retrieved pixel (Range 0 - 99).

```
Ocean Algorithm:
    High:    Good retrieval (uses only entries from TRMM apriori
             database)
    Medium:  Retrieval used extended database (created by lowering SST
             and Freezing level by 3K from TRMM observations)  and/or
             expanded search radius beyond 2K in SST and 3 mm in TPW
    Low:     Retrieval used excessive search radius to find matches in
             apriori database
Land/Coast Algorithm:
    High:    Good retrieval
    Medium:  Not currently used
    Low:     Pixel is ambiguous (Tb depression due to precipitation or
             surface effect)

Valid values include:
    0 : High quality (retrieval is good)
    1 : Medium quality (use with caution)
```

```
   2 : Low quality (recommended qualitative use only)
 -99 : Missing value
```

**pixelStatus** (1-byte integer, array size: npixel x nscan):
If there is no retrieval at a given pixel, pixelStatus explains the reason (Range 0 - 99).

```
   0 : Valid pixel
   1 : Boundary error in landmask
   2 : Boundary error in sea-ice check
   3 : Boundary error in sea surface temperature
   4 : Invalid time
   5 : Invalid latitude/longitude
   6 : Invalid brightness temperature
   7 : Invalid sea surface temperature
   8 : No retrieval due to sea-ice over water
   9 : No retrieval due to sea-ice over coast
  10 : Land/coast screens not able to be applied
  11 : Failure in ocean rain - no match with database profile Tbs
 -99 : Missing value
```

**surfaceType** (1-byte integer, array size: npixel x nscan):
Indicates the type of surface (Range 0 - 99).

```
  10 : Ocean
  11 : Sea ice
  12 : Partial sea ice
  20 : Land
  30 : Coast
 -99 : Missing value
```

**landAmbiguousFlag** (1-byte integer, array size: npixel x nscan):
Defines codes for uncertain/ambiguous retrievals over land (Range 0 - 99). Valid values are:

```
   0 : No information
  13 : Ambiguous T22V / 2 different scattering screens
  14 : Cannot discriminate precip from cold surface
  63 : Light precipitation
  64 : Cold surface
  65 : Grody light precipitation
  66 : Huffman ambiguous
 -99 : Missing value
```

**landScreenFlag** (1-byte integer, array size: npixel x nscan):
Diagnostic codes for rainfall screens over land (Range 0 - 99). Valid values are:

```
  0 : No information
-31 : Land retrieval found ice likely
-41 : Land retrieval found large polarization
      difference due to ice or sand
-51 : Warm 85H and Low 22V, or clear ocean likely in coast retrieval
-61 : Probable coastline in coast retrieval
-99 : Missing value
```

**oceanExtendedDbase** (1-byte integer, array size: npixel x nscan):
Percent of the extended database entries (i.e., beyond the TRMM database) used in the retrieval (Range 0 - 100). Valid values are:

```
  0 : Only the TRMM database entries are used
  N : N% of the entries from the extended database are used
100 : Only the extended database entries are used
-99 : Missing value
```

**oceanSearchRadius** (1-byte integer, array size: npixel x nscan):
Expansion of the search radius of the apriori database beyond the initial SST and TPW search range. The profiles for the rain_ocean procedure are grouped by SST and TPW. The individual pixels TPW and SST are used to retrieve a group of pixels from the database. If there are fewer than 1000 profile clusters found, the search radius is expanded. (Range 0 - 99). Valid values are:

```
  0 : Default search radius used
  1 : Search radius expanded by +/- 1 mm in TPW and +/- 1 degree in SST
  N : Search radius expanded by +/- N mm in TPW and +/- N degrees in SST
-99 : Missing value
```

**chiSquared** (2-byte integer, array size: npixel x nscan):
Error diagnostic for Optimal Estimation calculation of TPW and wind speed. Values greater than the number of channels (9 for TMI) should be considered suspect, with values greater than 18 of limited use. Rainfall is possible above these values. Values could range from 0 to 10000, but should be less than 100. The missing value is -9999.9.

**probabilityOfPrecip** (1-byte integer, array size: npixel x nscan):
A diagnostic variable, in percent, defining the fraction of raining vs. non-raining Dbase profiles that make up the final solution. Values range from 0 to 100 percent. Special values are defined as:
-99 Missing value

**sunGlintAngle** (1-byte integer, array size: npixel x nscan):
Conceptually, the angle between the sun and the instrument view direction as reflected off

the Earth's surface. More specifically, define a Sun Vector from the viewed pixel location on the earth ellipsiod-model surface to the sun. Also define an Inverse Satellite Vector from the pixel to the satellite. Then reflect the Inverse Satellite Vector off the earth's surface at the pixel location to form the Reflected Satellite View Vector. sunGlintAngle is the angular separation between the Reflected Satellite View Vector and the Sun Vector. When sunGlintAngle is zero, the instrument views the center of the specular (mirror-like) sun reflection. Values range from 0 to 180 degrees. Special values are defined as:
-99 Missing value

**freezingHeight** (2-byte integer, array size: npixel x nscan):
The height, in meters, of the $0^{o}$C isotherm above the earth ellipsoid. The missing value is -9999.

**surfacePrecipitation** (4-byte float, array size: npixel x nscan):
The instantaneous precipitation rate at the surface for each pixel. Check pixelStatus for a valid retrieval. Values are in mm/hr. Special values are defined as:
-9999.9 Missing value

**convectPrecipitation** (4-byte float, array size: npixel x nscan):
The instantaneous convective precipitation rate at the surface for each pixel. Check pixelStatus for a valid retrieval. Values are in mm/hr. Special values are defined as:
-9999.9 Missing value

**surfaceRain** (4-byte float, array size: npixel x nscan):
The instantaneous rain rate (liquid portion of precipitation) at the surface for each pixel. Check pixelStatus for a valid retrieval. Values are in mm/hr. Special values are defined as:
-9999.9 Missing value

**cloudWaterPath** (4-byte float, array size: npixel x nscan):
Total cloud liquid water in the column. Values range from 0 to 3000 $kg/m^2$. Special values are defined as:
-9999.9 Missing value

**rainWaterPath** (4-byte float, array size: npixel x nscan):
Total rain water in the column. Values range from 0 to 3000 $kg/m^2$. Special values are defined as:
-9999.9 Missing value

**iceWaterPath** (4-byte float, array size: npixel x nscan):
Total of all ice species (including cloud ice and precipitation ice) in the column. Values range from 0 to 3000 $kg/m^2$. Special values are defined as:
-9999.9 Missing value

**seaSurfaceTemperature** (4-byte float, array size: npixel x nscan):
Sea surface temperature. Values in degrees K. The missing value is -9999.9.

**totalPrecipitableWater** (4-byte float, array size: npixel x nscan):
Liquid equivalent of the total water vapor column. Values range from 0 to 75 mm. Special

values are defined as:
 -9999.9  Missing value

**windSpeed** (4-byte float, array size: npixel x nscan):
Wind speed at the sea surface. Values in m/s, 20m above the surface. The missing value is -9999.9.

**freezingHeightIndex** (1-byte integer, array size: npixel x nscan):
Freezing Height Index in the cluster array. See description below clusterScale. Values range from 1 to 13. Special values are defined as:
 -99  Missing value

**clusterNumber** (1-byte integer, array size: nspecies x npixel x nscan):
Cluster Number in the cluster array. See clusterScale description below clusterScale. Values range from 1 to 100. Special values are defined as:
 -99  Missing value

**clusterScale** (4-byte float, array size: nspecies x npixel x nscan):
clusterScale is used to scale the values of the cluster array.

```
In order to recover values in a profile use
the clusterNumber, clusterScale and freezingHeightIndex
parameters and select your species and level:

Where:
  L = profile level (1-18) Top of each level
      specified in HgtLayerTop
  S = species(1-6)
      1 = cloud water content
      2 = rain water content
      3 = cloud ice content
      4 = snow water content
      5 = graupel water content
      6 = latent heat
  F = freezingHeightIndex
  C = clusterNumber

In a Fortran program,
  Profile Value = clusterScale * cluster(S,F,L,C)

In a C program,
  Profile Value = clusterScale * cluster[C-1][L-1][F-1][S-1]
```

## C Structure Header file:

```
#ifndef _TK_2A12_H_
```

```
#define _TK_2A12_H_

#ifndef _L2A12_NAVIGATION_
#define _L2A12_NAVIGATION_

typedef struct {
    float scPosX;
    float scPosY;
    float scPosZ;
    float scVelX;
    float scVelY;
    float scVelZ;
    float scLat;
    float scLon;
    float scAlt;
    float scAttRoll;
    float scAttPitch;
    float scAttYaw;
    float SensorOrientationMatrix[3][3];
    float greenHourAng;
} L2A12_NAVIGATION;

#endif

#ifndef _L2A12_SCANSTATUS_
#define _L2A12_SCANSTATUS_

typedef struct {
    signed char missing;
    signed char validity;
    signed char qac;
    signed char geoQuality;
    signed char dataQuality;
    short SCorientation;
    signed char acsMode;
    signed char yawUpStat;
    signed char tmiIsStatus;
    double FractionalGranuleNumber;
} L2A12_SCANSTATUS;

#endif

#ifndef _L2A12_SCANTIME_
```

```
#define _L2A12_SCANTIME_

typedef struct {
    short Year;
    signed char Month;
    signed char DayOfMonth;
    signed char Hour;
    signed char Minute;
    signed char Second;
    short MilliSecond;
    short DayOfYear;
} L2A12_SCANTIME;

#endif

#ifndef _L2A12_SWATH_
#define _L2A12_SWATH_

typedef struct {
    L2A12_SCANTIME ScanTime;
    float Latitude[208];
    float Longitude[208];
    L2A12_SCANSTATUS scanStatus;
    L2A12_NAVIGATION navigation;
    signed char qualityFlag[208];
    signed char pixelStatus[208];
    signed char surfaceType[208];
    signed char landAmbiguousFlag[208];
    signed char landScreenFlag[208];
    signed char oceanExtendedDbase[208];
    signed char oceanSearchRadius[208];
    short chiSquared[208];
    signed char probabilityOfPrecip[208];
    signed char sunGlintAngle[208];
    short freezingHeight[208];
    float surfacePrecipitation[208];
    float convectPrecipitation[208];
    float surfaceRain[208];
    float cloudWaterPath[208];
    float rainWaterPath[208];
    float iceWaterPath[208];
    float seaSurfaceTemperature[208];
    float totalPrecipitableWater[208];
```

```
    float windSpeed[208];
    signed char freezingHeightIndex[208];
    signed char clusterNumber[208][6];
    float clusterScale[208][6];
} L2A12_SWATH;

#endif

#ifndef _L2A12_DATAHEADER_
#define _L2A12_DATAHEADER_

typedef struct {
    float heightLayerTop[28];
    float cluster[100][28][13][6];
} L2A12_DATAHEADER;

#endif

#endif
```

## Fortran Structure Header file:

```
STRUCTURE /L2A12_NAVIGATION/
    REAL*4 scPosX
    REAL*4 scPosY
    REAL*4 scPosZ
    REAL*4 scVelX
    REAL*4 scVelY
    REAL*4 scVelZ
    REAL*4 scLat
    REAL*4 scLon
    REAL*4 scAlt
    REAL*4 scAttRoll
    REAL*4 scAttPitch
    REAL*4 scAttYaw
    REAL*4 SensorOrientationMatrix(3,3)
    REAL*4 greenHourAng
END STRUCTURE

STRUCTURE /L2A12_SCANSTATUS/
    BYTE missing
    BYTE validity
    BYTE qac
```

```
    BYTE geoQuality
    BYTE dataQuality
    INTEGER*2 SCorientation
    BYTE acsMode
    BYTE yawUpStat
    BYTE tmiIsStatus
    REAL*8 FractionalGranuleNumber
END STRUCTURE

STRUCTURE /L2A12_SCANTIME/
    INTEGER*2 Year
    BYTE Month
    BYTE DayOfMonth
    BYTE Hour
    BYTE Minute
    BYTE Second
    INTEGER*2 MilliSecond
    INTEGER*2 DayOfYear
END STRUCTURE

STRUCTURE /L2A12_SWATH/
    RECORD /L2A12_SCANTIME/ ScanTime
    REAL*4 Latitude(208)
    REAL*4 Longitude(208)
    RECORD /L2A12_SCANSTATUS/ scanStatus
    RECORD /L2A12_NAVIGATION/ navigation
    BYTE qualityFlag(208)
    BYTE pixelStatus(208)
    BYTE surfaceType(208)
    BYTE landAmbiguousFlag(208)
    BYTE landScreenFlag(208)
    BYTE oceanExtendedDbase(208)
    BYTE oceanSearchRadius(208)
    INTEGER*2 chiSquared(208)
    BYTE probabilityOfPrecip(208)
    BYTE sunGlintAngle(208)
    INTEGER*2 freezingHeight(208)
    REAL*4 surfacePrecipitation(208)
    REAL*4 convectPrecipitation(208)
    REAL*4 surfaceRain(208)
    REAL*4 cloudWaterPath(208)
    REAL*4 rainWaterPath(208)
    REAL*4 iceWaterPath(208)
```

```
    REAL*4 seaSurfaceTemperature(208)
    REAL*4 totalPrecipitableWater(208)
    REAL*4 windSpeed(208)
    BYTE freezingHeightIndex(208)
    BYTE clusterNumber(6,208)
    REAL*4 clusterScale(6,208)
END STRUCTURE

STRUCTURE /L2A12_DATAHEADER/
    REAL*4 heightLayerTop(28)
    REAL*4 cluster(6,13,28,100)
END STRUCTURE
```